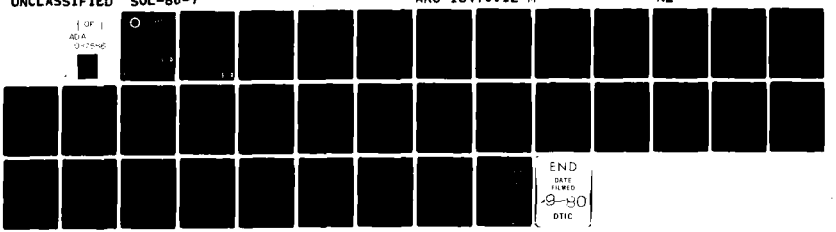
AD-A087 556 STANFORD UNIV CA SYSTEMS OPTIMIZATION LAB F/G 12/1
ASPECTS OF MATHEMATICAL MODELING RELATED TO OPTIMIZATION,(U)
MAY 80 P E GILL, W MURRAY, M A SAUNDERS DAAG29-79-C-0110
UNCLASSIFIED SOL-80-7 ARO-16470.12-M NL



END
DATE FILMED
9-80
DTIC

ARO 16470.12-M

LEVEL II

Systems
Optimization
Laboratory

ADA087556

DTIC
ELECTE
AUG 4 1980
S D
D

DDC FILE COPY

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

Department of Operations Research
Stanford University
Stanford, CA 94305

80 8 1 062

SYSTEMS OPTIMIZATION LABORATORY
DEPARTMENT OF OPERATIONS RESEARCH
STANFORD UNIVERSITY
STANFORD, CALIFORNIA 94305

ASPECTS OF MATHEMATICAL MODELING RELATED TO OPTIMIZATION

by

Philip E. Gill, Walter Murray,
Michael A. Saunders and Margaret H. Wright

TECHNICAL REPORT SOL 80-7
May 1980

Research and reproduction of this report were supported by the Department of Energy Contract DE-AC03-76SF00326, PA No. DE-AT03-76ER72018; National Science Foundation Grants MCS-7926009 and ENG77-06761; and the U.S. Army Research Office Contract DAAG29-79-C-0110.

ASPECTS OF MATHEMATICAL MODELING RELATED TO OPTIMIZATION †

Philip E. Gill, Walter Murray, Michael A. Saunders and Margaret H. Wright
Systems Optimization Laboratory
Department of Operations Research
Stanford University
Stanford, California 94305

ABSTRACT

Many practical optimization problems involve mathematical models of complex real-world phenomena. This paper discusses some aspects of modeling that influence the performance of optimization methods. Information and advice are given concerning the construction of smooth models, the transformation of an optimization problem from one category to another, scaling, formulation of constraints, and techniques for special types of models.

† An earlier version of this paper was presented at the conference "Software for Numerical Optimization", London, March 1978.

1. Introduction

Mathematical models are frequently used to study real-world phenomena that are not susceptible to analytic techniques alone, and to investigate the relationships among the parameters that affect the functioning of complex processes. Models provide an effective — sometimes, the only — means of evaluating the results of alternative choices; for example, a model is essential in cases where experimentation with the real-world system is prohibitively expensive, dangerous, or even impossible.

Optimization methods play an important role in modeling, because a model is not usually developed as an end in itself. Rather, the model is formulated in order to determine values of free parameters that produce an optimum measure of "goodness" — for instance, the most stable structure, or the best performance on observed data.

The relationship between the formulation of a model and the associated optimization can take several forms. In many instances, virtually all the effort of model development is aimed toward constructing a model that reflects the real world as closely as possible. Only after the form of the model is essentially complete is some thought given to a method for finding optimal values of the parameters. However, selection of an off-the-peg algorithm without considering properties of the model often leads to unnecessary failure or gross inefficiency.

On the other hand, we do not advocate over-simplification or distortion in formulation simply in order to be able to solve the eventual optimization problem more easily. There has been a tendency, particularly in the large-scale area, to model even highly nonlinear processes as linear programs, because until recently no nonlinear methods were available for very large problems. The effort to remove nonlinearities often leads to greatly increased problem size, and also significantly affects the nature of the optimal solution (e.g., a linear programming solution is always an extreme point of the feasible region, but the solution of a nonlinear program is usually not).

A model to be optimized should be developed by striking a reasonable balance between the aims of improved accuracy in the model (which usually implies added complexity in the formulation) and increased ease of optimization. This might be achieved by invoking an optimization procedure on successively more complicated versions of the model, in a form of "stepwise" refinement. Thus, the effects of each refinement in the model on the optimization process can be monitored, and fundamental difficulties can be discovered much more quickly than if no optimization were applied until the model was essentially complete. This is especially important when dealing with models that contain many interconnected sub-systems, each requiring extensive calculation.

This paper is not primarily concerned with how accurately models reflect the real world, but rather with aspects of modeling that influence the perfor-

mance of optimization algorithms. In particular, we shall discuss considerations in formulating models that contribute to the success of optimization methods. Our observations of practical optimization problems have indicated that, even with the best available software, the efficient optimization of a model can be critically dependent on certain properties of the formulation. It is often the case that the formulator of the model must make numerous arbitrary decisions that do not affect the accuracy of the model, yet are crucial to whether the model is amenable to solution by an optimization algorithm.

2. Classification of Optimization Problems

The most general form of an optimization problem is that of minimizing a scalar function of the independent variables (the *objective function*), subject to restrictions or *constraints* on acceptable values of the variables. We shall primarily be concerned with problems in which the set of acceptable variables is defined by relations involving continuous functions of the variables:

$$\text{NLP} \qquad \min_{x \in \mathbb{R}^n} \quad F(x)$$

$$\text{subject to} \quad c_i(x) = 0, \quad i = 1, 2, \ldots, m_1;$$
$$c_i(x) \geq 0, \quad i = m_1 + 1, \ldots, m_2.$$

In this formulation, the functions F and $\{c_i\}$ are termed the *problem functions*.

Constraints on the parameters may take other forms — e.g., some of the variables may be restricted to a finite set of values only. Problems of this type are generally much more difficult to solve than those of the form NLP; some possible approaches to models with such constraints are noted in Section 5.

An important point to be considered in modeling is whether the formulation has features that enhance ease of optimization, since a general algorithm for NLP will generally be inefficient if applied to a problem with special features. For purposes of choosing an algorithm, optimization problems are usually divided into categories defined by properties of the problem functions, where problems in each category are best solved by a different algorithm.

The following table gives a typical classification scheme, where significant advantage can be taken of each characteristic:

Properties of $F(x)$	Properties of $\{c_i(x)\}$
Linear	None
Sums of squares of linear functions	Simple bounds
Quadratic	Linear
Sums of squares of nonlinear functions	Sparse linear
Nonlinear	Nonlinear

Certain problem characteristics have a much greater impact on ease of optimization than others — for instance, consider problem size. Beyond one-dimensional problems (which are invariably treated as a special case), the next dividing line occurs when the problem size becomes so large that: (a) the data cannot all be stored in the working memory of the computer; (b) exploiting the sparsity (proportion of zeros) in the problem data leads to a significant improvement in efficiency. Before that point, however, the effort required to solve a typical problem is, roughly speaking, bounded by a reasonably behaved polynomial function of problem size. Therefore, increasing the number of parameters in an unconstrained problem from, say, 9 to 12 is usually not significant.

By contrast, the form of the problem constraints can have an enormous effect on the ease of solution. In particular, there is generally a very small increase (or possibly even a reduction) in difficulty when moving from an unconstrained problem to one with simple bounds on the variables; in fact, the optimization library from the National Physical Laboratory, England, solves unconstrained problems by calling a bound-constrained subroutine. General linearly constrained problems are noticeably more difficult to solve than those with bound constraints only, and the presence of nonlinear constraints introduces an even larger increase in difficulty. For this reason, it is sometimes advisable to reformulate a model so as to eliminate nonlinear constraints; this topic will be discussed further in Section 4.

Probably the most fundamental property of the problem functions with respect to ease of optimization is *differentiability*, which is important because algorithms are based on using available information about a function at one point to deduce its behavior at other points. If the problem functions are twice continuously differentiable, say, the ability of an algorithm to locate the solution is greatly enhanced compared to the case when the problem functions are non-differentiable. Therefore, most optimization software is designed to solve smooth problems, and there is a great incentive to formulate differentiable model functions (see Section 3). For a smooth problem within a specific category, there still remains a great deal of choice in algorithm selection, depending, for example, on how much derivative information is available, the relative cost of computing certain quantities, and so on. As a general rule, algorithms tend to become more successful and robust as more information is provided.

3. Avoiding unnecessary discontinuities

The word "unnecessary" appears in the title of this section because, strictly speaking, no function is continuous when evaluated with limited precision. Since only a finite set of numbers can be represented with a standard floating-point format, the usual mathematical definition of continuity, which involves arbitrarily

small perturbations in the function and its arguments, is not applicable. In general, the *computed* version of any function is inherently discontinuous. Fortunately, for a well-scaled function, the discontinuities can be regarded as insignificant, in that they do not adversely affect the performance of optimization methods that assume smoothness; however, poor scaling can lead to difficulties. The topic of problem scaling will be briefly discussed in Section 5.

Since optimization problems with general non-differentiable functions are difficult to solve, it is highly desirable for the user to formulate smooth mathematical models; problems with structured discontinuities will be discussed in Section 4.3. Before discussing means of avoiding non-differentiability, we stress that there is a crucial distinction between a function that is non-differentiable and a function whose derivatives are (for some reason) not computable. If a function is truly non-differentiable, its derivatives simply do not exist mathematically at all points — e.g., the function $\max(x_1, x_2)$ is in general non-differentiable when $x_1 = x_2$. By contrast, a function may be smooth, but its derivatives are not available because, say, of the complexity or expense of computing them; nonetheless, an algorithm may rely on their existence.

Careful consideration of the underlying mathematical model can often indicate whether a given function should be differentiable. If there are critical points in the *real-world process* — for example, a reservoir overflows, or an activity shifts from one resource to another — there will probably be discontinuities in the derivatives. If the user is uncertain about differentiability, little will usually be lost by assuming that the derivatives are continuous. If the chosen optimization algorithm subsequently fails, the user may switch to an algorithm for non-smooth functions.

3.1 The role of accuracy in model functions.

A common fallacy arises when only a limited accuracy is required in the optimal solution of a modeling problem (for example, when the model formulation is known to neglect significant elements in the real-world process, or the model function represents an observed phenomenon whose form is deduced from data of limited accuracy). In such an instance, the modeler may believe that the problem functions need to be evaluated to only slightly more than the required number of significant figures during optimization.

Because the real-world function, say $F_R(x)$, is only approximated by an ideal mathematical model function, say $F_M(x)$, the user is essentially assuming that an optimization method will tolerate convenient changes in the representation of $F_M(x)$ that are smaller in magnitude than the known accuracy to which F_M approximates F_R. However, this assumption is *not* warranted if the changes introduce serious discontinuities into the model function or its derivatives, or cause other substantive deviations in the nature of the model function. Let ϵ_M

denote the percentage error in $F_M(x)$ due to fundamental deficiencies in the model. This error will not in general be known precisely, but often a lower bound can be estimated from, say, the accuracy of the data or the significance of neglected processes. If ϵ_M is very small, the modeler will tend to exercise the appropriate care in the computer implementation of F_M, in order to preserve the high accuracy. However, in a typical model ϵ_M lies in the range 0.1%–5.0%. In this case, suppose that there are two possible computable approximations of F_M, say F_A and F_B, which can also be considered as functions that approximate F_R. The functions F_A and F_B differ from the idealized model function F_M in that, for convenience of implementation and computation, an additional error, say ϵ_C, has been introduced; however, ϵ_C is guaranteed to be much smaller than ϵ_M — say, $\epsilon_C \approx .01\%$. Since this error does not significantly increase the existing error in approximating F_R, the three approximations F_M, F_A, and F_B could be considered of equal merit in one sense — their closeness in value to F_R.

Consider the specific example in one dimension illustrated in Figures 1*a* and 1*b* (the errors in the figures have been exaggerated to emphasize the aspect of interest). If the errors $|F_R(x)-F_A(x)|$ and $|F_R(x)-F_B(x)|$ were the sole concern, then the two approximations F_A and F_B would be equally good.

With respect to use by an optimization method, however, F_A and F_B are quite different. In particular, F_B has the same smoothness properties as the underlying (unknown) F_R, whereas F_A has discontinuities in both function and derivatives at many points. The derivative discontinuities alone would have several bad effects on an optimization method. First, the method might well converge to a spurious local minimum of F_A. Another harmful result of using F_A would occur within algorithms that approximate derivatives by finite differences. If the small step of the finite-difference interval happened to cross a derivative discontinuity, the approximation of the gradient would be completely inaccurate, even if the gradient were well-defined at the current point.

It may seem that these cautionary remarks would apply only to a small number of uninformed people, since presumably no one would deliberately include significant discontinuities in the modeling function or its derivatives. Although discontinuities at the level of round-off error are inevitable in any model, unacceptably large discontinuities are sometimes introduced by modelers who assume that other "minor" changes are of no significance.

3.2 Approximation by series or table look-up.

In our experience, one of the most common causes of lack of smoothness is the occurrence of a discontinuity in the evaluation of some subsidiary function, say $W(\gamma)$, upon which $F(x)$ depends. Since computers can perform only elementary arithmetic operations, more complicated functions are approximated in various ways — often by a truncated series expansion, the choice of which depends on

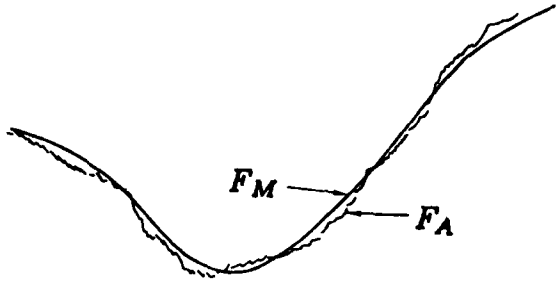


Figure 1*a*. A discontinuous approximation to a smooth function.

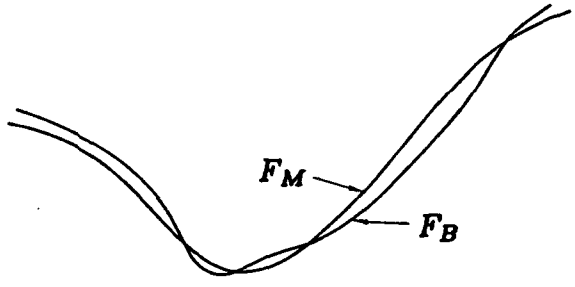


Figure 1*b*. A continuous approximation to a smooth function.

the argument. Thus, it may happen that $W(\gamma)$ is evaluated using two formulae, one for small $|\gamma|$ and another for large $|\gamma|$. Although both should give the identical result at the crossover point, in general the truncation error will be different for the two series. Even if only a single series is used, discontinuities may occur because the evaluation process includes more terms of the series for certain values of the argument — e.g., at $\gamma = 4.7$ four terms of the series are used, whereas at $\gamma = 4.7 + 10^{-15}$ five terms are used.

To avoid such discontinuites (or at least minimise their effect), the user is advised to do the following:

(i) whenever possible, avoid switches in formulae (for example, by using a fixed number of terms in representing a function by an approximating series);

(ii) if there is a switch, ensure that the function values (and, *whenever possible*, the first derivatives) match at the cross-over point;

Unfortunately, switches in formulae sometimes occur without the user's knowledge. For example, a standard software library routine for evaluating the Bessel function $J_0(\gamma)$ uses two different methods, depending on whether γ is

greater than 11. In such a case, the user may be required to utilize an alternative procedure for evaluating the subsidiary function.

A related way in which discontinuites are introduced is by including a "table look-up" during the computation of the model function. Suppose that $F(x)$ depends on the quantity $V(\gamma)$, and that $V(\gamma)$ is tabulated for the set of values $\gamma = 0(0.01)1$. If $V(.6243)$ is required, the user may believe that $V(0.62)$ is an entirely adequate approximation. Although this might be true in some cases (as discussed in Section 3.1 with F_R and F_A), this treatment would make $V(\gamma)$ a piecewise constant function, with undesirable discontinuities if its properties are reflected in F. Linear interpolation within the table will produce continuity in $V(\gamma)$ (and hence, usually in F), but it will still produce discontinuities in the first derivatives. The best solution — which is always realizable — is to avoid tables completely, and to replace them by smooth approximating functions such as splines. Even two-way tables (those that require two parameters) can now be adequately represented by smooth surfaces (see Hayes, 1970; Powell, 1977).

3.3 Sub-problems based on iteration.

A more subtle source of discontinuities can be observed when evaluation of a function contains sub-problems — for example, a system of differential equations or an integral. The solution of these sub-problems to full machine precision (even if possible) generally requires considerable computational effort, and thus tends to be regarded as unwarranted by the modeler, since the integral, differential equation, or whatever, is only an approximation to some more complicated real-world phenomenon. A frequent example is the unconstrained minimization with respect to x of the integral

$$F(x) = \int_a^b f(x,t)dt. \tag{1}$$

Typically, the function $f(x,t)$ cannot be integrated analytically. Hence, a numerical quadrature scheme must be used, in which the integral is approximated by a weighted sum of function values at selected points:

$$\int_a^b f(x,t)dt \approx I(t,\omega) \equiv \sum_{j=1}^{M} \omega_j f(x,t_j), \tag{2}$$

where $\{\omega_j\}$ are the weights and $\{t_j\}$ are a set of abscissae such that $a \leq t_1 \leq \ldots \leq t_M \leq b$. The error in the approximation (2) depends on the higher derivatives of f, the number of abscissae $\{t_j\}$, and the position of $\{t_j\}$ within $[a,b]$ (Dahlquist and Björck, 1974).

Among the most efficient methods for numerical integration are the *adaptive* quadrature techniques, in which the abscissae in (2) are chosen dynamically, based on the sampled behavior of f during an iterative procedure; the idea is to place more points in regions where f appears to be less well-behaved. Several good software packages are available for adaptive quadrature, and the user may well have chosen one of these state-of-the-art codes for evaluating the function (1). However, unless the integrals are evaluated to full machine precision, the function (1) may not be "well-behaved" in all necessary senses. In the case of evaluating (1), use of an adaptive quadrature technique will tend to cause the same unfortunate consequences noted earlier with series representation. In particular, the inherently iterative nature of adaptive quadrature means that widely varying numbers of points may be placed in different parts of $[a,b]$ for very close values of x. Although a similar accuracy will generally be attained in the approximate integral for all values of x in $[a,b]$, the model function tends to contain undesirable discontinuities. Therefore, the curve of the approximate integral computed by an adaptive quadrature technique may well resemble that of F_A in Figure 1*a*.

It should be stressed that adaptive quadrature is inappropriate only because the sub-problem that it enters is part of an outer problem in which smoothness is more important than accuracy, at least far from the solution.

An alternative way to proceed is to devise a fixed (smooth) quadrature formula I (as in (2)) to be used as input to the optimization routine, and thereby to determine $\bar{x}$, the point at which I achieves its minimum. It would be fortuitous indeed if $\bar{x}$ were an acceptable approximation to $\overset{*}{x}$, the minimum of (2), and therefore another step in the procedure is carried out. For instance, a more accurate quadrature formula (say, involving substantially more terms) can be devised, and the optimization process repeated, using $\bar{x}$ as the starting point; if $f(x,t)$ is well-behaved, a judicious choice of abscissae may allow a better estimate of the integral without unduly increasing the number of points. Since $\bar{x}$ should be a reasonably close approximation to $\overset{*}{x}$, only a relatively small number of evaluations of the more complex quadrature formula should be required. If a highly accurate integral at $\overset{*}{x}$ is the ultimate aim, the final step could be application of an adaptive quadrature technique at the single point $\overset{*}{x}$. This example illustrates that it is often worthwhile to interleave modeling and optimization, since the creation of increasingly accurate quadrature formulae for smaller intervals is in fact a modeling process.

4. Problem transformation

4.1 Simplifying or eliminating constraints.

In the past, algorithms for unconstrained optimization were more numerous and more effective than those for constrained problems. Today, however, algorithms for problems with only simple bounds or linear constraints are comparable in efficiency to unconstrained algorithms. Therefore, it is virtually never worthwhile to transform bound-constrained problems (in fact, it is often beneficial to *add* bounds on the variables), and it is rarely appropriate to alter linearly constrained problems.

Any problem transformation should be undertaken only with extreme care. In particular, some "folklore" transformations may cause an increase in problem difficulty, and may not even produce the desired result. For example, to solve

$$\begin{aligned} &\underset{w \in \mathbb{R}^n}{\text{minimize}} && F(w) \\ &\text{subject to} && w_i \geq 0, \end{aligned}$$

it is *not* sufficient to minimize $\mathcal{F}(z)$ with $w_i = z_i^2$. To see why not, consider the case when $F(w) = w^{5/2}$.

Furthermore, it is inadvisable to replace a problem with inequality constraints of the form $c_i(x) \geq 0$ by one with equality constraints that include squared extra variables, i.e., $c_i(x) - y_i^2 = 0$. If it is considered necessary to eliminate inequalities in this manner, a preferable transformation is to add a slack variable whose non-negativity is imposed with a bound: $c_i(x) - y_i = 0$; with $y_i \geq 0$.

Nonetheless, transformation to an unconstrained problem or a problem with simple constraints can be an effective method of allowing the model to be solved more easily. This can sometimes be achieved simply by judicious choice of the model's independent variables. In any transformation, it is important to ensure that the new problem is not more difficult than the original one. Certain transformations of the variables may lead to the following difficulties:

(i) the desired minimum may be inadvertently excluded;

(ii) the degree of nonlinearity may be significantly increased;

(iii) the scaling may be adversely affected;

(iv) the new function may contain singularities not present in the original problem;

(v) the Hessian matrix may become singular or ill-conditioned in the region of interest;

(vi) the transformed problem may have additional local minima;

(vii) the function may be periodic in the new variables.

It is not easy to formulate general rules that will avoid these problems. In our experience, however, trigonometric and exponential transformations tend as a class to create more numerical difficulties than alternative approaches, especially as the number of variables increases.

The problem of periodicity can be offset to some extent in two ways. First, an unconstrained algorithm can be modified as follows. Suppose that the transformed variables are $\{y_i\}$, and that

$$F(y + ja_ie_i) = F(y), \quad j = \pm 1, \pm 2, \ldots$$

If the step to be taken in y_i is p_i, p_i should be altered by adding or subtracting a multiple of a_i until $|p_i| < a_i$. A second way to avoid difficulties with periodicity is to impose simple bounds on the appropriate variables — e.g., if x_1 represents an angle, add to the problem statement the requirement that $0 \leq x_1 \leq 2\pi$, and use a bound-constrained algorithm.

We shall illustrate other difficulties that may occur because of problem transformation by the following example:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \; F(x) \tag{3a}$$

$$\text{subject to} \sum_{i=1}^{n} x_i^2 = 1. \tag{3b}$$

If we make the transformation

$$\begin{aligned} x_1 &= \sin y_1 \cdots \sin y_{n-1}, \\ x_i &= \cos y_{i-1} \sin y_i \cdots \sin y_{n-1}, \quad i = 2, \ldots, n, \end{aligned}$$

the problem becomes

$$\underset{y \in \mathbb{R}^{n-1}}{\text{minimize}} \; \mathcal{F}(y). \tag{4}$$

Some additional difficulties have been introduced into the transformed problem (4). In addition to the obvious periodicity, the new function is invariant to changes in any of the first $n-2$ variables if y_{n-1} is zero. Furthermore, if any y_i, $i > 1$, is close to zero, $\mathcal{F}(y)$ is almost invariant to changes in the other variables; clearly, the problem has become very badly scaled.

An alternative transformation to satisfy automatically the constraint (3b) is to define the new y variables via:

$$a = \pm\left(1 + \sum_{i=1}^{n-1} y_i^2\right)^{\frac{1}{2}} \tag{5}$$

$$x_i = y_i/a, \quad i = 1, \ldots, n-1 \tag{6}$$

$$x_n = 1/a. \tag{7}$$

The new problem is then

$$\text{minimize } \min(F_P(y), F_N(y)),$$

where $F_P(y)$ is the function obtained by choosing the plus sign in (5) and substituting for x in $F(x)$, with an analogous definition of F_N.

In practice, if the sign of any of the optimal x_i is known, that variable could become the one whose value is fixed by (7), thereby removing the need to define two functions. It is preferable to choose an x_i whose value is not close to zero, since in this case some of the other transformed variables would become badly scaled. Beware also that if x_i were subject to certain bounds, e.g. $0.1 \leq x_i \leq 0.2$, it would not be safe in general to eliminate that variable.

Despite their drawbacks, transformations involving trigonometric expressions are desirable in some situations. For example, consider a problem in which the variables are the coordinates in three dimensions of a set of k points, which are constrained to lie on the surface of a sphere. The problem in this form is then

$$\begin{aligned} &\underset{x,y,z\in\mathbb{R}^k}{\text{minimize}} && F(x,y,z) \\ &\text{subject to} && x_i^2 + y_i^2 + z_i^2 = r^2, \quad i = 1,2,\ldots,k. \end{aligned} \tag{8}$$

Note that there are $3k$ variables and k constraints.

In general, a proper elimination of t equality constraints from a problem with n unknowns leads to an unconstrained problem with $n - t$ variables. For this example, a trigonometric representation of the variables allows the constraints of (8) to be satisfied automatically, by introducing a set of $2k$ angles $\{\theta_i, \psi_i\}$, which become the new variables, such that

$$\begin{aligned} x_i &= r \sin\theta_i \cos\psi_i \\ y_i &= r \sin\theta_i \sin\psi_i \\ z_i &= r \cos\theta_i. \end{aligned}$$

To avoid difficulties with the periodic nature of the function, simple bounds can be imposed:

$$\begin{aligned} 0 \leq \theta_i &\leq 2\pi \\ 0 \leq \psi_i &\leq 2\pi. \end{aligned}$$

In fact, it may be possible to make these bounds more restrictive to ensure some topological property of the set of points. In general, upper and lower bounds should be *as close as possible*.

Alternatively, the points might be restricted to lie within a sphere of radius r. In this case, k additional variables $\{d_i\}$ could be added, where d_i gives the distance of the i-th point from the origin, and satisfies the simple bound $d_i \leq r$. The constraints on the set of points would then become

$$x_i^2 + y_i^2 + z_i^2 + d_i^2 = r^2,$$

and the definitions of $\{\theta_i, \psi_i\}$ would be altered accordingly.

Although in general it is not worthwhile to eliminate only some, but not all, nonlinear constraints, this rule does not apply to sparse problems. When the sparsity of the constraints is significant, it is beneficial to replace constraints that involve a large number of variables by constraints in which only a small number of variables appear.

A device frequently used by engineers to convert a problem with nonlinear equality constraints into an unconstrained problem is to add a "Lagrangian" term to the objective function. This practice is based on the following result: under mild conditions, the solution $\overset{*}{x}$ of the problem

$$\begin{aligned} &\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad F(x) \\ &\text{subject to} \quad c_i(x) = 0, \quad i = 1, 2, \ldots, t. \end{aligned} \tag{9}$$

is a stationary point of the Lagrangian function $F(x) - \lambda^T c(x)$, where λ satisfies

$$\nabla F(\overset{*}{x}) = \sum_{i=1}^{t} \lambda_i \nabla c_i(\overset{*}{x}). \tag{10}$$

This property can be used to formulate either an unconstrained problem in the x variables only (combined with some procedure for estimating λ), or a system of nonlinear equations involving both x and λ. A complete treatment of this topic is beyond the scope of this discussion (see, for example, Powell, 1974, or Wright, 1979, for details). However, the point to be stressed here is that this transformation is *not* guaranteed to be successful. The solution of (9) is in general a stationary point, not a minimum, of the Lagrangian function, although obviously it is a minimum in some cases. Furthermore, since the transformation preserves the solution only with the vector λ that satisfies (10), a poor estimate of λ may cause divergence. If one attempts to estimate x and λ simultaneously with an $(n + t)$-dimensional nonlinear system, the matrix involved is often extremely ill-conditioned or even singular. Because of these and other subtle difficulties that can occur, the transformation carries a serious risk of failure. Since several algorithms for nonlinearly constrained optimization are based on using the same result, a user will have a much better chance of success by applying a soundly implemented Lagrangian-based algorithm designed specifically to handle constrained problems.

4.2 Problems where the variables are continuous functions.
An important class of problems that are not immediately expressible in the finite-dimensional form NLP involves optimization with respect to a specified set of *functions*. For example, the problem may be to compute the minimum of the

integral

$$\int_a^b f(x(t),t)\,dt \tag{11}$$

for a given f, over all smooth functions $x(t)$ defined on $[a,b]$. In many instances, such a problem can be "solved" as a finite-dimensional problem; we shall illustrate the idea of the transformation through a detailed treatment of (11).

Since the functional form of $x(t)$ cannot be obtained in general, it is necessary to represent $x(t)$ by a finite amount of information. Clearly it would be infeasible to store the finite, but enormous, set of values of $x(t)$ at each machine-representable point in the interval. Instead, we must be content with storing a reasonable amount of information, from which a satisfactory approximation to $x(t)$ can be constructed. This usually involves little sacrifice because the desired result in most practical problems is simply a compact representation of the behavior of $x(t)$ — typically, the values of $x(t)$ at a set of points in $[a,b]$. This set of information can be interpreted as an implicit definition of a new function $\tilde{x}(t)$, obtained by applying some form of interpolation to approximate the value of $x(t)$ at non-tabulated points. The accuracy of $\tilde{x}(t)$ depends on the smoothness of x and $\tilde{x}$, the number and placement of the interpolating points, etc. (see Dahlquist and Björck, 1974).

A satisfactory solution to the original problem (11) is then a representation of $\tilde{x}(t)$. Let

$$\tilde{x}(t) = \sum_{j=1}^{q} c_j w_j(t), \tag{12}$$

where $\{c_j\}$ are a set of coefficients and $\{w_j(t)\}$ are a set of known basis functions. Examples of frequently used basis functions are: (i) polynomials: $w_j = t^{j-1}$; (ii) Chebyshev polynomials: $w_j = T_{j-1}(t)$; and (iii) B-splines: $w_j = M_j(t)$ (see Hayes, 1970; Cox, 1977).

If the form (12) for $\tilde{x}$ is substituted for x in the objective function, the infinite-dimensional problem becomes a finite-dimensional problem with unknowns $\{c_j\}$. Depending on the nature of f, the integral (11) can then be computed analytically or from a quadrature rule; see Section 3.3 for further comments on the use of quadrature rules.

4.3 Transformation of composite non-differentiable functions.

Although non-differentiable problems are in general more difficult to solve, a distinction must be made between a problem with random discontinuities in functions or derivatives, and one in which a great deal of information is known about the nature of any discontinuities. In the latter case, algorithms can take advantage of the special structure.

In some well-known instances, the problem functions themselves are not smooth, but rather are composites of smooth functions. For example, the following non-differentiable functions frequently occur in models, and are composed in a particular way from the set of smooth functions $\{f_i\}$:

(a) $F = \max(f_1, f_2, \ldots, f_m)$;
(b) $F = \sum_{i=1}^{m} |f_i|$;
(c) $F = \sum_{i=1}^{m} \max(f_i, 0)$.

There has been much research concerning effective methods for these and related problems, and it is therefore advisable to use a specialized algorithm (see Wolfe, 1975; Murray and Overton, 1979). However, if such an algorithm is not available, a composite non-differentiable problem can sometimes be transformed into a smooth, but more complex, problem. To illustrate this type of transformation, we consider three common composite problems.

Problem 1:

$$\text{minimize } \max(f_1(x), f_2(x), \ldots, f_m(x)), \tag{13}$$

where $\{f_i(x)\}$ are smooth functions. This problem can be transformed into a smooth problem by introducing a new variable x_{n+1}, which is an upper bound on all the functions $\{f(x)_i\}$. The new problem is then

$$\begin{aligned} &\text{minimize} && x_{n+1} \\ &\text{subject to} && f(x)_i \leq x_{n+1}, \quad i = 1, 2, \ldots, m. \end{aligned}$$

Note that the original unconstrained problem has been transformed into a nonlinearly constrained problem — the reverse effect from the transformations considered in Section 3. In fact, all transformations of non-differentiable composite functions lead to a similar increase in complexity.

Problem 2:

$$\text{minimize } \sum_{i=1}^{m} |f_i(x)|. \tag{14}$$

To transform (14), we note that a typical function $f_i(x)$ may be split into its positive and negative parts by writing $f_i(x) = r_i - s_i$, where both r_i and s_i are non-negative. The relationship $|f_i(x)| \leq r_i + s_i$ leads to the following transformation of (14):

$$\begin{aligned} &\text{minimize} && \sum_{i=1}^{m} r_i + s_i \\ &\text{subject to} && f_i(x) = r_i - s_i, \quad i = 1, 2, \ldots, m, \\ & && r_i \geq 0; \ \ s_i \geq 0, \quad i = 1, 2, \ldots, m. \end{aligned}$$

Problem 3:

$$\text{minimize} \sum_{i=1}^{m} \max(f_i(x), 0). \tag{15}$$

A similar argument to that used in transforming (14) gives

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{m} r_i \\ \text{subject to} \quad & f_i(x) = r_i - s_i, \quad i = 1, 2, \ldots, m, \\ & r_i \geq 0; \quad s_i \geq 0, \quad i = 1, 2, \ldots, m. \end{aligned}$$

Although there is a significant increase in the number of variables with the latter two approaches, this need not be a serious obstacle to the indicated transformations if an algorithm is used that exploits bounds.

Problems 1 and 2 often arise in the context of data fitting, in which case $F(\overset{*}{x})$ is expected to be small. If $F(\overset{*}{x})$ is actually zero, then in either case the problem is equivalent to solving the smooth nonlinear least-squares problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \sum_{i=1}^{m} f_i^2(x).$$

If $F(\overset{*}{x})$ is small (although not zero), an alternative to solving Problems 1 or 2 is to solve a sequence of weighted least-squares problems of the form

$$\text{minimize} \sum_{i=1}^{m} w_i^{(k)} f_i^2, \qquad k = 1, 2, 3, \ldots. \tag{16}$$

Let $\overset{*}{x}^{(k)}$ denote the solution of (16), and let $w_i^{(1)} = 1/m, i = 1, \ldots, m$.

To solve Problem 1, the weights in the sequence of problems are chosen to be

$$w_i^{(k+1)} = \frac{1}{S} f_i^2(\overset{*}{x}^{(k)})\, w_i^{(k)},$$

where $S = \sum f_i^2(\overset{*}{x}^{(k)})\, w_i^{(k)}$, and is chosen to make $\sum w_i^{(k+1)} = 1$. For Problem 2, the sequence of weights is defined as

$$w_i^{(k+1)} = \frac{S}{|f_i(\overset{*}{x}^{(k)})|},$$

where $S = \sum 1 / (|f_i(\overset{*}{x}^{(k)})|)$. If, say, t values of $f_i(\overset{*}{x}^{(k)})$ are zero, the corresponding elements of $w^{(k+1)}$ are set to $1/t$, and the rest to zero. Usually, only two or three least-squares problems of the form (16) must be solved to obtain convergence.

In either case, it is not necessary to compute $\overset{*}{x}^{(k)}$ to high accuracy. Since $\overset{*}{x}^{(k)}$ is used as the initial point when solving for $\overset{*}{x}^{(k+1)}$, only a few iterations are typically required to find the solution of (16) for $k > 1$. See Lawson and Hanson, 1974, for a more complete discussion of these techniques.

5. Scaling

The term "scaling" is invariably used in a vague sense to discuss numerical difficulties whose existence is universally acknowledged, but cannot be described precisely in general terms. Therefore, it is not surprising that much confusion exists about scaling, and that authors tend to avoid all but its most elementary aspects.

The discussion of scaling in this paper will be restricted to simple transformations of the variables, and special techniques in nonlinear least-squares problems. A much more complete discussion of scaling is given in Gill *et al.* (1980), which includes suggestions for improving the scaling of constrained as well as unconstrained problems.

5.1 Scaling by transformation of variables.

Scaling by variable transformation converts the variables from units that typically reflect the physical nature of the problem to units that display certain desirable properties during the minimization process.

There is an important distinction between transforming variables to improve the behaviour of an optimization method and transforming variables to change the problem category; the latter type of transformation is discussed in Section 4.1.

The first basic rule of scaling is that the variables of the scaled problem should be of similar magnitude and of order unity in the region of interest. Within optimization routines, convergence tolerances and other criteria are necessarily based upon an implicit definition of "small" and "large", and thus variables with widely varying orders of magnitude may cause difficulties for some algorithms. If typical values of all the variables are known, a problem can be transformed so that the variables are all of the same order of magnitude, as illustrated in the following example. Consider a problem that involves a gas/water heat exchanger. Table 1 gives the variables, their interpretation, and a typical value for each.

Table 1 — Typical values of unscaled variables

Variable	Interpretation	Units	Typical Value
x_1	Gas flow	lbs/hour	11,000
x_2	Water flow	lbs/hour	1,675
x_3	Steam thermal resistance	$(\text{BTU/(hour ft}^2\,^\circ\text{F)})^{-1}$	100
x_4	Waste build-up	$(\text{BTU/(hour ft}^2\,^\circ\text{F)})^{-1}$	6×10^{-4}
x_5	Gas-side radiation	$\text{BTU/(hour ft}^2\,^\circ\text{R}^4\text{)}$	5.4×10^{-10}

The magnitudes of the variables arise simply from the units in which they are expressed. Since most of the variables are measured in terms of different physical units, there is no reason to suppose that they will be of similar size (in fact, the variables in the table are obviously of enormously different magnitudes). Even when the physical units of measure are the same, there may be a marked difference in typical values — for example, there is a difference between the physical properties of water and waste product.

Normally, only linear transformations of the variables should be used to rescale (although occasionally nonlinear transformations are possible). The most commonly used transformation is of the form

$$x = Dy,$$

where $\{x_j\}$ are the original variables, $\{y_j\}$ are the transformed variables, and D is a constant diagonal matrix.

For the variables given in Table 1, an adequate scaling procedure would be to set d_j, the j-th diagonal element of D, to a typical value of the j-th variable. For instance, d_1 could be set to 1.1×10^4.

Unfortunately, this simple type of transformation has the disadvantage that some accuracy may be lost, as the following example illustrates. Suppose that a variable x_j is known to lie in the range [200.1242, 200.1806]. If the variable is scaled by the "typical value" 200.1242, the scaled variables will lie in the range [1.0, 1.000282] (to seven significant figures). On a computer with seven decimal digits of precision, only the three least significant figures are available to represent the variation in y_j, and consequently four figures of accuracy are lost whether the scaling is performed or not.

Another disadvantage of scaling by a diagonal matrix only is that the magnitude of a variable may vary substantially during the minimization. In this event, what might be a good scaling at one point may prove harmful at another.

Both of these disadvantages can be overcome if we know a realistic *range* of values that a variable is likely to assume during the minimization. For example, such a range may be provided by simple upper and lower bound constraints that have been imposed upon the variables. Suppose that the variable x_j will always lie in the range $a_j \leq x_j \leq b_j$. A new variable y_j can be defined as

$$y_j = \frac{2x_j}{b_j - a_j} - \frac{a_j + b_j}{b_j - a_j}. \tag{17}$$

The transformation (17) can be written in matrix form as

$$x = Dy + c, \tag{18}$$

where D is a diagonal matrix with j-th diagonal element $(b_j - a_j)/2$, and c is a vector with j-th element $(a_j + b_j)/2$. This transformation guarantees that

$-1 \leq y_j \leq +1$ for all j, regardless of the value of x_j within the interval $[a_j, b_j]$. In the example noted above, the appropriate transformation (17) for the variable in the range [200.1242, 200.1806] is

$$x_j = 0.0282 y_j + 200.1524,$$

which allows y_j to be represented to full precision within the range $[-1, +1]$.

We emphasize that the interval specifying the range of values for a given variable must be a realistic one. Under no circumstances should this type of transformation be used when the value of a_j or b_j is simply a crude limit, possibly wrong by several orders of magnitude.

When the variables are scaled by a linear transformation of the form (18), the derivatives of the objective function are also scaled. Let g_y and G_y denote the gradient vector and Hessian matrix of the transformed problem; the derivatives of the original and transformed problems are then related by

$$g_y = Dg; \quad G_y = DGD.$$

Hence, even a "mild" scaling such as $x_j = 10 y_j$ may have a substantial effect on the Hessian, and this in turn may significantly alter the convergence rate of an optimization algorithm.

5.2 Scaling nonlinear least-squares problems.

Nonlinear least-squares problems most commonly arise when a model function, say $y(x, t)$, needs to be fitted as closely as possible to the set of observations $\{y_j\}$ at the points $\{t_j\}$. The important feature of nonlinear data-fitting problems is that the variables to be estimated can sometimes be scaled automatically by scaling the independent variable t.

To see how this may happen, we consider the following example. The formulation is a simplified version of a real problem, but the original names of the variables have been retained. The function to be minimized is

$$\sum_{j=1}^{m} \left(\frac{y(p_j) - Y_j}{\Delta y_j} \right)^2,$$

where p is the independent variable, which lies in the range [566, 576], and the data points $\{Y_j\}$ and their associated errors $\{\Delta y_j\}$ are given. The functional form assumed for $y(p)$ is

$$y(p) = \sum_{j=0}^{J} A_j p^j + \sum_{k=1}^{K} B_k \exp\left(-\frac{(p - p_k)^2}{2\sigma_k^2} \right), \tag{19}$$

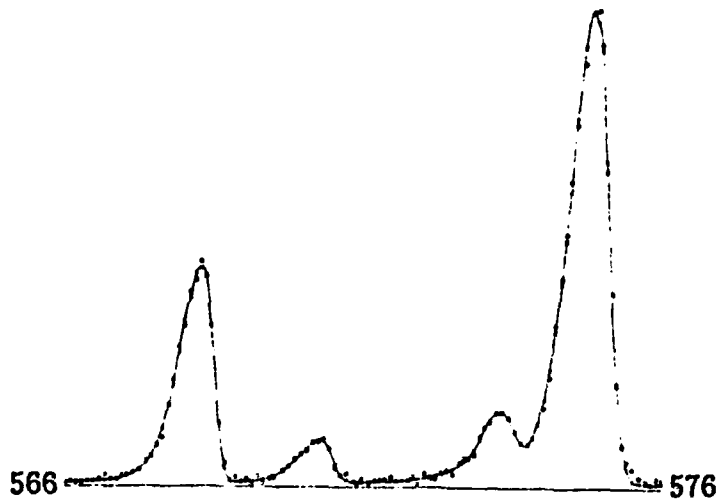


Figure 2. Typical data set and fitting function.

where the parameters to be estimated are $\{A_j\}$, $j = 1, \ldots, J$, and $\{B_k, p_k, \sigma_k\}$, $k = 1, \ldots, K$. A typical data set and fitting function $y(p)$ are shown in Figure 2.

The problem can be interpreted as fitting a Gaussian curve to each of the K peaks, together with a background function (the first term on the right-hand side of (19)). For the example depicted in Figure 2, K is four, and $\{p_k\}$, $k = 1, \ldots, 4$, are estimates of the corresponding peak positions; clearly each p_k lies in the range [566, 576].

The major difficulty with solving this problem is that, even for moderate j, A_j must be very small because of the size of p_j. For example, if $j = 3$, A_3 is multiplied in (19) by at least $566^3 \approx 10^8$. Scaling the independent variable p so that each A_j lies in the range $[-1, +1]$ partially solves the problem. This can be done by defining a new independent variable z such that $p = 576z$. However, this transformation has the same disadvantages noted in the previous section for a purely diagonal scaling, namely, that relative precision in z is lost unnecessarily. However, since a meaningful range of y values is known exactly, the transformation

$$p = 5z + 571$$

may be used. With this transformation, both z and the transformed values of A_j are in the range $[-1, +1]$, and no relative precision is lost in the values of z.

The transformed function is then

$$\Phi(z) \equiv \sum_{j=1}^{J} \bar{A}_j z^j + \sum_{k=1}^{K} B_k \exp\left(\frac{-(z-\bar{z}_k)^2}{2\bar{\sigma}_k^2}\right).$$

Often it is not necessary to recompute $\{A_j\}$, $\{\bar{p}_k\}$, and $\{\sigma_k\}$ from $\{\bar{A}_j\}$, $\{\bar{z}_k\}$, and $\{\bar{\sigma}_k\}$. For example, we may wish to compute the area under the $\Phi(z)$ curve, or to compute values of $y(p)$ at values other than $\{p_j\}$. In such cases the transformed function is just as useful as the original (and often much better).

6. Formulation of constraints

6.1 Indeterminacy in constraint formulation.

A difficulty in formulating a model with constraints on the variables is the possibility of creating a poorly posed optimization problem, even though the underlying model has a well defined solution. This situation can exist for many reasons, which are too numerous to list here. For example, redundant constraints may be included that are simply linear combinations of other constraints, in order to provide a summary of certain activities. Such features may serve a useful purpose within the model, and the modeler knows that they "should" have no effect on the optimal values of the model parameters. Unfortunately, the performance of optimization algorithms may thereby be adversely affected.

A typical situation occurs when the variables in the optimization problem do not correspond directly to the model parameters. As an illustration of such a model, we briefly mention a problem posed by Professor Alice Whittemore of Stanford University. Her work involved a statistical model of data concerning the incidence of lung cancer. The model variables were three-way probabilities $\{p_{ijk}\}$, $i = 1,\ldots,I$; $j = 1,\ldots,J$; $k = 1,\ldots,K$, and the objective function depended only on $\{p_{ijk}\}$. In one of the models considered, it was assumed that these probabilities could be represented as the product of three two-way probabilities, i.e.

$$p_{ijk} = f_{ij}\, g_{jk}\, h_{ik}. \tag{20}$$

Further constraints were also imposed upon $\{f_{ij}\}$, $\{g_{jk}\}$, and $\{h_{ik}\}$, which then served as the variables in a nonlinearly constrained optimization problem.

The presence of indeterminacy in the optimization problem was revealed when the solution method consistently experienced great difficulty in converging, which was surprising in view of the quadratic convergence that would usually be expected. In order to discover the source of the difficulty, several different

starting points were tried. The method converged to completely different values of $\{f_{ij}\}$, $\{g_{jk}\}$, and $\{h_{ik}\}$, but always yielded the same values for all $\{p_{ijk}\}$ and the optimal objective function. This behavior led to a re-examination of the model formulation, which showed that the problem variables were not uniquely defined by the optimization problem. Examination of (20) shows that the value of p_{ijk} is unaltered if f_{ij} and g_{jk}, say, are replaced by αf_{ij} and g_{jk}/α for any $\alpha \neq 0$. Such a change did not affect satisfaction of the remaining constraints, and hence the problem contained an inherent indeterminacy with respect to the chosen variables. In fact, if a less robust algorithm had been used to solve the nonlinearly constrained problem, it would have failed to converge, since the linear systems to be solved for the search direction were exactly singular. In this case, the lack of uniqueness was easily resolved by imposing some additional normalization constraints on one of the sets of variables — e.g., $\sum_k g_{jk} = 1$.

This example is not particularly unusual, and highlights the importance of applying modeling and optimization interactively. Although similar difficulties may be less simple to diagnose and correct, the general rule of thumb is to check that the solution of the optimization problem is as well defined as the underlying model.

6.2 The use of tolerance constraints.

Equality constraints occur in problem formulations for a variety of reasons. Often the very nature of the variables imposes an equality constraint — for example, if the variables $\{x_i\}$ represent proportions or probabilities, this gives rise to the constraint $\sum_{i=1}^{n} x_i = 1$ (as well as non-negativity restrictions). Constraints of this type are "genuine" equalities, in the sense that the computed solution must satisfy them exactly (where "exactly" means "within working precision"). However, it is not unusual in modeling that constraints that might seem initially to be firm equality constraints should be treated instead as constraints that need not be satisfied with maximum possible accuracy. For example, this situation occurs when the underlying model is known to contain inaccuracies. The term *tolerance constraint* refers to a range constraint with a very narrow range, which gives the effect of satisfying an equality constraint only to within a prescribed tolerance. Thus, the linear constraint

$$a^T x = b$$

would be replaced by

$$b - \epsilon_2 \leq a^T x \leq b + \epsilon_1, \tag{21}$$

where ϵ_1 and ϵ_2 are small, but not negligible, positive quantities (exactly the same transformation can be made for a nonlinear constraint).

Tolerance constraints of the type (21) differ from ordinary range constraints because the range of acceptable values, although non-zero, is very small. In some problems, treating constraints of this type as equalities may cause there to be no feasible solution, or may distort the properties of the solution if the corresponding constraints are ill-conditioned (e.g., points that satisfy the constraints exactly may be far removed from other points that lie within the given range).

The following detailed example illustrates this situation for both linear and nonlinear constraints, and also includes other forms of problem transformation. The statement of the problem has been simplified in order to highlight the features of interest. The model is to be used in the design of a platinum catalyst converter that controls the exhaust emission on car engines. The corresponding optimization problem was originally posed in terms of a set of equations modeling the chemical reactions within the converter. There are two types of equations: nonlinear equations that describe the reaction rates for various chemical processes, and linear relationships that arise from the conservation of the constituent elements. In total, there are eight variables and thirteen equations (eight nonlinear and five linear).

The variables $\{x_1, \ldots, x_8\}$ represent the partial pressures of the following species, in the order given: propane, carbon monoxide, nitrogen oxide, carbon dioxide, oxygen, water, nitrogen, and hydrogen. Clearly it is required that $x_i \geq 0$, $i = 1, \ldots, 8$, since negative values would have no physical meaning. The eight nonlinear reaction equations are as follows, where the constants $\{K_1, \ldots, K_8\}$ are the reaction constants whose logarithms are defined by logarithms in the temperature:

$$\frac{x_4^3 x_6^4 x_7^5}{x_1 x_3^{10}} - K_1 = f_1(x) = 0 \tag{22}$$

$$\frac{x_4^3 x_6^4}{x_1 x_5^5} - K_2 = f_2(x) = 0 \tag{23}$$

$$\frac{x_4^3 x_8^{10}}{x_1 x_6^6} - K_3 = f_3(x) = 0 \tag{24}$$

$$\frac{x_4 \sqrt{x_7}}{x_2 x_3} - K_4 = f_4(x) = 0 \tag{25}$$

$$\frac{x_4 x_8}{x_2 x_6} - K_5 = f_5(x) = 0 \tag{26}$$

$$\frac{x_6\sqrt{x_7}}{x_3x_8} - K_6 = f_6(x) = 0 \tag{27}$$

$$\frac{x_6}{x_3\sqrt{x_5}} - K_7 = f_7(x) = 0 \tag{28}$$

$$\frac{x_4}{x_2\sqrt{x_5}} - K_8 = f_8(x) = 0. \tag{29}$$

The linear equations derived from conservation of elements are the following, where the constants $\{a_1, \ldots, a_8\}$ represent the initial partial pressures of the various species:

Oxygen balance

$$x_2 + x_3 + 2x_4 + 2x_5 + x_6 - a_2 - a_3 - 2a_4 - 2a_5 - a_6 = f_9(x) = 0 \tag{30}$$

Carbon balance

$$3x_1 + x_2 + x_4 - 3a_1 - a_2 - a_4 = f_{10}(x) = 0 \tag{31}$$

Hydrogen balance

$$8x_1 + 2x_6 + x_8 - 8a_1 - 2a_6 - a_8 = f_{11}(x) = 0 \tag{32}$$

Nitrogen balance

$$x_3 + 2x_7 - a_3 - 2a_7 = f_{12}(x) = 0 \tag{33}$$

Total balance

$$\sum_{i=1}^{8} x_i - \sum_{i=1}^{8} a_i = f_{13}(x) = 0. \tag{34}$$

The usual method for solving a set of overdetermined equations is to minimize the sum of squares of the residuals, i.e.

$$\text{minimize} \sum_{i=1}^{13} f_i^2(x). \tag{35}$$

One difficulty with this approach is that the equations (22) - (34) are of two distinct types, and it is desirable to preserve the natural separation of linear and nonlinear equations during the process of solution. A means of allowing the latter is to include only the nonlinear equations in the sum of squares, and to ...he problem as

$$\text{minimise} \sum_{i=1}^{8} f_i^2(x) \tag{36}$$

subject to the five linear equality constraints (30) – (34). If the problem is formulated as (36), the computed solution will satisfy the linear constraints exactly. In this situation, however, representation of the real-world phenomena by equality constraints may be undesirable, since it is known that not all possible chemical reactions have been included in the conservation equations (as many as thirty processes involving only minute quantities were omitted from the formulation). Therefore, forcing equality upon imprecise constraints may be imprudent. In fact, in some instances there would be no feasible solution for this model because of the additional non-negativity constraints. To avoid this difficulty, the equality constraints (30) – (34) might be replaced by tolerance constraints. The selection of each tolerance can be judged from the percentage of each reaction process that is dominated by the main reaction (the one represented in the constraints).

For some models, this adjustment of the constraints would suffice to allow the problem to be solved satisfactorily. However, in this instance poor scaling causes additional difficulties, since the reaction constants $\{K_i\}$ vary enormously (for instance, K_1 is of order 10^{250}). One method for overcoming poor scaling here is to replace each $f_i(x)$, $i = 1, \ldots, 8$ by the transformed function

$$F_i = \ln(f_i(x) + K_i) - \ln K_i,$$

and then to minimise the new objective function

$$\sum_{i=1}^{8} F_i^2(x).$$

Although such a transformation cures the difficulty due to the variation in magnitude of $\{K_i\}$, another indication of poor scaling is the extreme sensitivity of the solution to differences in parameter values that would ordinarily be considered negligible. For example, the functions vary dramatically depending on whether x_1, is 10^{-14} or 10^{-100}, whereas standard computer algorithms would undoubtedly treat these quantities as equivalent. To overcome this difficulty, a *nonlinear* transformation of the variables is necessary, and therefore any suitable transformation destroys the linearity of the constraints (30) – (34). Fortunately, for this problem there is a nonlinear transformation that changes the nonlinear functions F_i into linear functions, namely

$$x_i = e^{y_i}. \tag{37}$$

Note that the transformation (37) also ensures that $x_i \geq 0$. To illustrate the effect on the nonlinear functions, consider F_6, which is transformed to

$$\hat{F}_6(y) = \frac{1}{2}y_7 + y_6 - y_3 - y_8 - \ln K_6.$$

The effect on the linear equations is illustrated by equation (32), which becomes

$$\hat{f}_{11}(y) = 8e^{y_1} + 2e^{y_2} + e^{y_3} - b_3 = 0.$$

With the transformation (37), we obtain the following linearly constrained problem:

$$\begin{aligned} &\underset{y}{\text{minimise}} \quad \sum_{i=9}^{13} \hat{f}_i^2(y) \\ &\text{subject to} \quad \hat{F}_i(y) = 0, \quad i = 1, 2, \ldots, 8. \end{aligned} \tag{38}$$

Note that (38) would not be a satisfactory representation if the original linear constraints were expected to be satisfied exactly, since in general we would not expect $\hat{f}_i(y)$ to be zero at the solution of (38).

Even (38) is still not satisfactory because the linear constraints of (38) (the transformed nonlinear constraints of the original problem) will generally be incompatible, solely because the values of $\{K_i\}$ have been determined from inherently imprecise experimental data. Hence the equality constraints of (38) should be replaced by tolerance constraints, where the tolerance for each constraint is determined by the relative accuracy of the corresponding K_i. It is interesting to note that if the $\{K_i\}$ are only slightly in error (as they should be), the system of equations defined by the constraints of (38) is only slightly incompatible. In an initial solution of the catalyst converter problem, the incompatibility was much larger than expected, and this revealed an error in the original data.

7. Problems with discrete or integer variables

Many practical problems occur in which some of the variables are restricted to be members of a finite set of values. These variables are termed *discrete* or *integer*. Examples of such variables are: items that are obtainable or manufactured in certain sizes only, such as the output rating of pumps or girder sizes; or the number of journeys made by a traveler. Such limitations mean that the standard definitions of differentiability and continuity are not applicable, and consequently numerical methods for differentiable problems must be used indirectly (except for a certain number of special cases where the solution of the continuous problem is known to satisfy the discrete/integer constraints automatically).

If the objective and constraint functions are linear, many special integer linear programming methods have been developed, notably variants of "branch and bound"; in some other special cases, dynamic programming methods can be applied (see Beale, 1977). However, we shall be concerned with mixed integer-nonlinear problems, i.e. nonlinear problems with a mixture of discrete and continuous variables.

It is important to distinguish between two types of discrete variables, since different methods can be applied to help solve each problem category. We shall illustrate the distinction, and possible approaches for dealing with such variables, by considering two typical problems in some detail.

5.1 Pseudo-discrete variables.

The first problem concerns the design of a network of urban sewer or drainage pipes. Within a given area, the position of a set of manholes is based on needs of access and the geography of the street layout. It is required to interconnect the manholes with straight pipes so that the liquid from a wide catchment area enters the manholes and flows under gravity down the system and out of the area. Each manhole has several input pipes and a single output pipe. To facilitate the flow, the pipes are set at an angle to the horizontal.

The variables of the problem are the diameters of the pipes and the angles that the pipes make with the horizontal. The constraints are: the slope of a given pipe lies between upper and lower bounds (determined by the need to facilitate flow and comply with the topography of the street level); the pipe diameters are non-decreasing as flow moves down the system; and the flow in the pipes (a nonlinear function of the pipe diameters and slopes) lies between some maximum and minimum values when the system is subjected to a specific "steady state" load. The objective of the design is to minimize the cost of construction of the pipe network while still providing an adequate level of extraction. The major costs in constructing the system are the costs of digging the trenches for the pipes and the capital costs of the pipes themselves. These costs are complementary, since narrow pipes are cheap, but require the excavation of deep sloping trenches to carry the required load.

At first sight the problem appears to be a straightforward nonlinearly constrained problem with continuous variables. What makes it a mixed continuous-discrete problem is the fact that pipes are manufactured only in standard diameters. The variables corresponding to the diameters of the pipes are examples of the first type of discrete variable, which occurs when the solution to the continuous problem (in which the variables are not subject to the discrete restrictions) is perfectly meaningful, but cannot be accepted due to extraneous restrictions. Such variables will be termed *pseudo-discrete*, and occur frequently in practice.

As we shall now demonstrate by the sewer-network example, problems with pseudo-discrete variables can often be solved by utilizing the solution of the continuous problem. This suggestion relies on the well behaved nature of the functions in most practical models — i.e., if the optimal pipe diameter when treated as a continuous variable is, say, 2.5723 feet, the optimal discrete value is unlikely to be very different.

In a general problem with pseudo-discrete variables, suppose that x_1 must assume one of the values $d_1, d_2, \ldots, d_r$. Let x^c denote the value of x at the solution of the continuous problem, which is assumed to be unique. Suppose that x_1^c satisfies

$$d_s < x_1^c < d_{s+1}.$$

The value of the objective function F at x^c is a lower bound on the value of F at any solution of the discrete problem, since if x_1 is restricted to be any value other than x_1^c, the objective function for such a value must be larger than $F(x^c)$, irrespective of the values of $x_2, \ldots, x_n$.

The next stage of the solution process is to fix the variable x_1^c at either d_s or d_{s+1} (usually, the nearer value); similarly, any other discrete variable may be set in this manner. The problem is then solved again, minimizing with respect to the remaining continuous variables, using the old optimal values as the initial estimate of the new solution. Solving the restricted problem should require only a fraction of the effort needed to solve the continuous problem, since the number of variables is smaller, and the solution of the restricted problem should be close to the solution of the continuous problem. The solution of the restricted problem, say x^r, is not necessarily optimal since incorrect values may have been selected for the discrete variables. Since $F(x^c)$ is a lower bound on $F(x^r)$, a value of $F(x^r)$ close to $F(x^c)$ will be a satisfactory solution in most practical problems. If it is thought worthwhile to seek a lower value than $F(x^r)$, some of the discrete variables may be set at their alternative values. Usually, these trials are worthwhile for those variables whose "continuous" value lies close to the centre of the range. Typically, very few such trials are necessary in practice.

In some cases, the restriction of a discrete variable may have the beneficial effect of automatically narrowing the choice of others. For example, in the pipe network problem, pipes lower down the network cannot be smaller in diameter than those upstream. Consequently, setting x_1 to d_{s+1}, say, may fix the choice for x_2.

Discrete variables may also be chosen to achieve an overall balance in the value of $F(x)$. For example, if the problem concerns the selection of girder sizes for minimizing the weight of a bridge, some girder sizes could be increased to make the bridge take an increased load, but others could be simultaneously decreased to achieve only a small increase in overall weight.

It is important in such problems to note that the solution of the continuous problem can always be used as an initial estimate for the solution of a restricted problem. In many practical problems, only two or three solutions of a restricted problem are needed to determine an acceptable solution of a discrete problem. The extra computing cost in solving the additional restricted problems associated with the discrete variables is likely to be a fraction of the cost to solve the original full continuous problem; if not, this implies that the discrete solution differs

substantially from the continuous solution. In such circumstances, it may be worthwhile to alter the extraneous conditions so that the two solutions are closer. For example, in problems of supply, such as the urban sewer problem, alternative supplies may be sought whose specifications are closer to the corresponding elements of the continuous problem, since by definition this change would yield a significant reduction in construction costs.

5.2 Integer variables.

The second type of discrete variable is one for which there is no sensible interpretation of a non-integer value — for example, when the variable represents a number of items, or a switch from one mutually exclusive strategy or resource to another (e.g., the change between coating materials in lens design). This type of discrete variable problem is much more difficult to solve than the first. If the number of such variables is small, say less than five, and the number of distinct values that each variable can take is also small, a combinatorial approach is possible. In this context a combinatorial technique is one in which the objective function is minimized for every possible combination of values that the discrete variables can assume. It may happen in practice that some combinations are considered unlikely, and so not all cases need to be tried. A combinatorial approach is often reasonable for constrained problems because many infeasible combinations can be eliminated before any computation takes place. In addition, with a combinatorial approach there is a useable solution no matter where the algorithm is terminated.

Unfortunately, for larger numbers of variables the combinatorial approach becomes too expensive, as the number of possible cases grows extremely large very quickly. For some discrete-variable problems that arise in practice, it is possible to pose a related problem with only continuous variables, such that the solution of the new problem, although not identical to the solution of the original, serves as a guide to the likely values of the discrete variables.

We illustrate this approach by considering a simplified version of a problem solved by Professor R. H. Sargent, Imperial College, concerning the design of a distillation column; for further details, the reader is referred to Sargent and Gaminibandara, 1976. Figure 3 displays a simplified diagram of a distillation column. Vapour is introduced at the bottom of the column and flows upwards. Condensed vapour flows downwards and is recycled using a boiler. The column is divided into a number of stages, and at some of the stages additional liquid (known as feed) is introduced. At each stage the liquid and vapour mix and alter in composition. The liquid is then drawn off as a product, or is used as an input

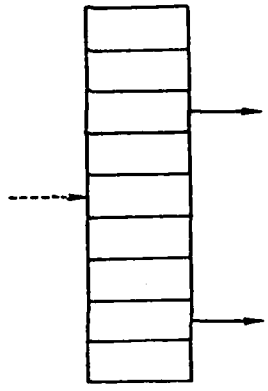


Figure 3. Diagram of a distillation column.

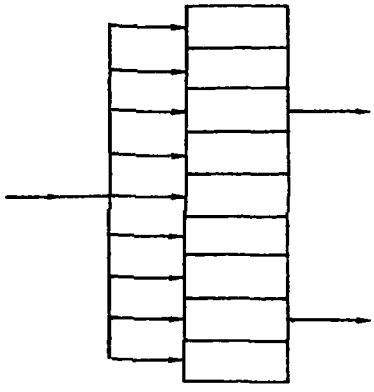


Figure 4. New model of a distillation column.

to the neighbouring stages. The optimization problem is to choose the level at which to place the feed input in order to achieve a specified performance at minimum cost.

At first sight it might be thought that the variable associated with the feed level has no continuous analogue. However, Sargent and Gaminibandara introduced a set of new variables, where each new variable corresponds to a stage of the column, and represents the percentage of the total feed to be input at that particular level. The new model is depicted in Figure 4. The problem is then re-formulated and solved, treating the new variables as continuous, and its solution is taken to indicate properties of the solution of the original problem. For example, if the solution of the continuous problem indicates that 90 percent of the feed should go to a particular stage, this stage is likely to be the one at which to input the feed in the discrete model. Figure 5 shows some typical

percentage feed levels for a 9-stage continuous model; stage four appears to be the most likely candidate for the value of the discrete variable.

It is interesting to note that the results of Sargent and Gaminibandara suggested that a change in the design of some distillation columns should be considered. In some cases, the continuous solution indicated that the feed should be added at two separated stages — a design that had previously not been considered.

In conjunction with this and similar schemes, a term can be introduced into the objective function that has the effect of encouraging a single choice for a discrete variable. For example, in the continuous model of the distillation column, a term like $1/\sum x_i^2$ might be added to the objective function, where x_i is the fraction of the total input to the i-th stage.

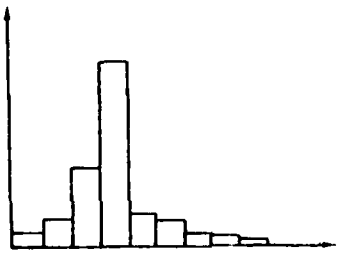


Figure 5. Typical optimal percentage feed levels.

8. Conclusions

We have described some standard (and, for the most part, straightforward) techniques that can make optimisation problems arising from modeling more amenable to solution by standard algorithms and software. Almost all of the methods given here have been used successfully in our own experiences with real problems. Certain cautionary guidelines have also been suggested in the hope of avoiding frequent pitfalls.

Of course, the nature of possible models varies so much that it is impossible to treat all relevant aspects of modeling. The main point of this paper is that developers of models should consider in the initial stages the ultimate need to solve an optimisation problem, since it is unlikely that optimization software will ever reach the state wherein a general routine can be used with impunity.

References

Beale, E. M. L. (1977). "Integer programming", in *The State of the Art in Numerical Analysis* (D. Jacobs, ed.), pp. 409–448, Academic Press, London and New York.

Cox, M. G. (1978). "A survey of numerical methods for data and function approximation", in *The State of the Art in Numerical Analysis* (D. Jacobs, ed.), pp. 627–668, Academic Press, London and New York.

Dahlquist, G. and Björck, A. (1974). *Numerical methods*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Gill, P. E., Murray, W., Saunders, M. A. and Wright, M. H. (1980). Automatic scaling for nonlinear optimization, Report (to appear), Department of Operations Research, Stanford University.

Hayes, J. G. (ed.) (1970). *Numerical approximation to functions and data*, Academic Press, London and New York.

Lawson, C. L. and Hanson, R. J. (1974). *Solving Least Squares Problems*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Murray, W. and Overton, M. L. (1979). A projected Lagrangian algorithm for nonlinear minimax optimization, Report SOL 79-21, Department of Operations Research, Stanford University.

Powell, M. J. D. (1974). "Introduction to constrained optimization", in *Numerical Methods for Constrained Optimisation* (P. E. Gill and W. Murray, eds.), pp. 1–28, Academic Press, London and New York.

Powell, M. J. D. (1977). "Numerical methods for fitting functions of two variables", in *The State of the Art in Numerical Analysis* (D. Jacobs, ed.), pp. 563–604, Academic Press, London and New York.

Sargent, R. W. H. and Gaminibandara, K. (1976). "Optimal design of plate distillation columns", in *Optimisation In Action* (L. C. W. Dixon, ed.), pp. 267–314, Academic Press, London and New York.

Wolfe, P. (ed.) (1975). Nondifferentiable Optimization, *Math. Prog. Study* 3.

Wright, M. H. (1979). Algorithms for nonlinearly constrained optimization, Report SQL 79-24, Department of Operations Research, Stanford University.

(9) (11) (14)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: SOL-80-7	2. GOVT ACCESSION NO.: AD-A087	3. RECIPIENT'S CATALOG NUMBER: 556
4. TITLE (and Subtitle): ASPECTS OF MATHEMATICAL MODELING RELATED TO OPTIMIZATION		5. TYPE OF REPORT & PERIOD COVERED: TECHNICAL REPORT; PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Philip E. Gill, Walter Murray, Michael A. Saunders, Margaret H. Wright		8. CONTRACT OR GRANT NUMBER(s): DAAG29-79-C-0110
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Department of Operations Research - SOL, Stanford University, Stanford, CA 94305		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: U.S. Army Research Office, P.O. Box 12211, Research Triangle Park, NC 27709		12. REPORT DATE: May 80; 13. NUMBER OF PAGES: 31
14. MONITORING AGENCY NAME & ADDRESS (If different from Controlling Office): DA-G29-79-C-0110, DE-AC03-76SF00326		15. SECURITY CLASS. (of this report): UNCLASSIFIED; 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

THE VIEW, OPINIONS, AND/OR FINDINGS CONTAINED IN THIS REPORT ARE THOSE OF THE AUTHOR(S) AND SHOULD NOT BE CONSTRUED AS AN OFFICIAL DEPARTMENT OF THE ARMY POSITION, POLICY, OR DECISION, UNLESS SO DESIGNATED BY OTHER DOCUMENTATION.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

model formulation
mathematical modeling
choice of optimization algorithms
problem transformation
transformation fo variables
non-standard problems
scaling

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Many practical optimization problems involve mathematical models of complex real-world phenomena. This paper discusses some aspects of modeling that influence the performance of optimization methods. Information and advice are given concerning the construction of smooth models, the transformation of an optimization problem from one category to another, scaling, formulation of constraints, and techniques for special types of models.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)